CALCULAODRA BINARIA

**Alumne:** Jordi Ribellas Ramos

**Professor:** Francis Real

**Curs:** 2022/23

# Enunciat

## En aquesta activitat es demana:

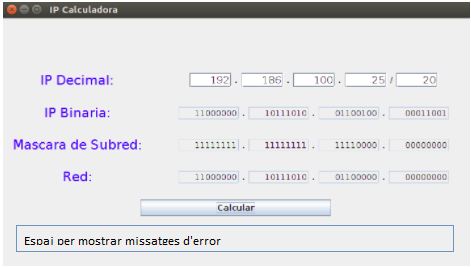
Crea una calculadora de IPs.

A partir de una IP decimal y de su mascara en formato reducido, calcula:

- La IP binaria

- La mascara binaria

- La IP de la red



- El campo inferior se utilizará para mostrar mensajes de error que puedan surgir, como por ejemplo si faltan datos, si hay algún número superior a 255, y los que se os ocurran.

- En el caso de que el usuario no informese la máscara de subred no se considerará error, sino que se entendrá que deberán aplicarse las máscaras de subred asociadas a las tipologías estándares A, B, C, D y E, teniendo en cuenta que en las dos últimas (D y E) no se debe informar ninguna máscara de subred.

# Codi

## En aquest apartat es veurà el meu codi.

**import** java**.**awt**.**BorderLayout**;**

**import** java**.**awt**.**Container**;**

**import** java**.**awt**.**FlowLayout**;**

**import** java**.**awt**.**GridLayout**;**

**import** java**.**awt**.**event**.**ActionEvent**;**

**import** java**.**awt**.**event**.**ActionListener**;**

**import** javax**.**swing**.**JButton**;**

**import** javax**.**swing**.**JFrame**;**

**import** javax**.**swing**.**JLabel**;**

**import** javax**.**swing**.**JPanel**;**

**import** javax**.**swing**.**JTextField**;**

**public** **class** CalculadoraBinaria **extends** JFrame **{**

**private** JTextField**[]** ipDecimal **=** **new** JTextField**[**5**];**

**private** JTextField ipMascara **=** **new** JTextField**(**3**);**

**private** JTextField**[]** ipBinari **=** **new** JTextField**[**5**];**

**private** JTextField**[]** ipMasBinari **=** **new** JTextField**[**5**];**

**private** JTextField**[]** xarxaBinari **=** **new** JTextField**[**5**];**

**private** JTextField Error **=** **new** JTextField**(**10**);**

**public** CalculadoraBinaria**()** **{**

**super(**"Calculadora IP Binaria"**);**

setDefaultCloseOperation**(**JFrame**.**EXIT\_ON\_CLOSE**);**

setSize**(**900**,** 300**);**

setTextIpDecimal**();**

setTextBi**();**

setTextMascara**();**

setTextXarxaBinari**();**

**// Decimal**

JPanel panelIP **=** **new** JPanel**();**

panelIP**.**setLayout**(new** FlowLayout**());**

panelIP**.**add**(**ipDecimal**[**1**]);**

panelIP**.**add**(new** JLabel**(**"."**));**

panelIP**.**add**(**ipDecimal**[**2**]);**

panelIP**.**add**(new** JLabel**(**"."**));**

panelIP**.**add**(**ipDecimal**[**3**]);**

panelIP**.**add**(new** JLabel**(**"."**));**

panelIP**.**add**(**ipDecimal**[**4**]);**

panelIP**.**add**(new** JLabel**(**"/"**));**

panelIP**.**add**(**ipMascara**);**

**// Binari**

JPanel panelBinari **=** **new** JPanel**();**

panelBinari**.**setLayout**(new** FlowLayout**());**

panelBinari**.**add**(**ipBinari**[**1**]);**

panelBinari**.**add**(new** JLabel**(**"."**));**

panelBinari**.**add**(**ipBinari**[**2**]);**

panelBinari**.**add**(new** JLabel**(**"."**));**

panelBinari**.**add**(**ipBinari**[**3**]);**

panelBinari**.**add**(new** JLabel**(**"."**));**

panelBinari**.**add**(**ipBinari**[**4**]);**

**// Mascara Binari**

JPanel panelSubred **=** **new** JPanel**();**

panelSubred**.**setLayout**(new** FlowLayout**());**

panelSubred**.**add**(**ipMasBinari**[**1**]);**

panelSubred**.**add**(new** JLabel**(**"."**));**

panelSubred**.**add**(**ipMasBinari**[**2**]);**

panelSubred**.**add**(new** JLabel**(**"."**));**

panelSubred**.**add**(**ipMasBinari**[**3**]);**

panelSubred**.**add**(new** JLabel**(**"."**));**

panelSubred**.**add**(**ipMasBinari**[**4**]);**

**// XarxaBinari**

JPanel panelRed **=** **new** JPanel**();**

panelRed**.**setLayout**(new** FlowLayout**());**

panelRed**.**add**(**xarxaBinari**[**1**]);**

panelRed**.**add**(new** JLabel**(**"."**));**

panelRed**.**add**(**xarxaBinari**[**2**]);**

panelRed**.**add**(new** JLabel**(**"."**));**

panelRed**.**add**(**xarxaBinari**[**3**]);**

panelRed**.**add**(new** JLabel**(**"."**));**

panelRed**.**add**(**xarxaBinari**[**4**]);**

**// Grid organizat**

JPanel panelDatos **=** **new** JPanel**();**

GridLayout gl **=** **new** GridLayout**(**5**,** 2**);**

panelDatos**.**setLayout**(**gl**);**

panelDatos**.**add**(new** JLabel**(**"IP Decimal: "**));**

panelDatos**.**add**(**panelIP**);**

panelDatos**.**add**(new** JLabel**(**"IP Binaria: "**));**

panelDatos**.**add**(**panelBinari**);**

panelDatos**.**add**(new** JLabel**(**"Mascara Subred: "**));**

panelDatos**.**add**(**panelSubred**);**

panelDatos**.**add**(new** JLabel**(**"Red: "**));**

panelDatos**.**add**(**panelRed**);**

Error**.**setEditable**(false);**

panelDatos**.**add**(**Error**);**

**// Boto de calcular**

JButton boton **=** **new** JButton**(**"Calcular"**);**

boton**.**setBounds**(**10**,** 10**,** 10**,** 10**);**

boton**.**addActionListener**(new** EventBoto**());**

**// Panell Sud**

JPanel panelSud **=** **new** JPanel**();**

panelSud**.**setLayout**(new** FlowLayout**());**

panelSud**.**add**(**boton**);**

Container cp **=** getContentPane**();**

cp**.**add**(**panelDatos**,** BorderLayout**.**CENTER**);**

cp**.**add**(**panelSud**,** BorderLayout**.**SOUTH**);**

**}**

**public** **class** EventBoto **implements** ActionListener **{**

@Override

**public** **void** actionPerformed**(**ActionEvent e**)** **{**

**int** a **=** 0**;**

**boolean** errors **=** **false;**

**int** mascara **=** 0**;**

**for** **(int** i **=** 1**;** i **<** ipDecimal**.**length**;** i**++)** **{**

**if** **(**ipDecimal**[**i**].**getText**().**isEmpty**()** **&&** ipMascara**.**getText**().**isEmpty**())** **{**

Error**.**setText**(**"Hi ha algun camp buit"**);**

errors **=** **true;**

**}** **else** **if** **(!**isNumero**(**ipDecimal**[**i**].**getText**().**toString**()))** **{**

Error**.**setText**(**"Algun camp de la ip no te numeros"**);**

errors **=** **true;**

**}else** **{**

a **=** Integer**.**parseInt**(**ipDecimal**[**i**].**getText**());**

mascara **=** Integer**.**parseInt**(**ipMascara**.**getText**());**

**}**

**if** **(**a **>** 255 **||** a **<** 0**)** **{**

Error**.**setText**(**"La Ip no es correcte"**);**

errors **=** **true;**

**}** **else** **if** **(**mascara **>** 32 **||** mascara **<** 0**)** **{**

Error**.**setText**(**"La mascara no es correcte"**);**

errors **=** **true;**

**}**

**}**

**if** **(**errors **==** **false)** **{**

calcularIP**();**

calcularMascara**();**

calcularXarxa**();**

Error**.**setText**(**""**);**

**}** **else** **{**

**for** **(int** i **=** 1**;** i **<** 5**;** i**++)** **{**

ipBinari**[**i**].**setText**(**""**);**

ipMasBinari**[**i**].**setText**(**""**);**

xarxaBinari**[**i**].**setText**(**""**);**

**}**

**}**

**}**

**}**

**public** **void** calcularIP**()** **{**

**int** a**,** operador**;**

String intString **=** ""**,** provisional **=** ""**;**

**for** **(int** i **=** 1**;** i **<** ipDecimal**.**length**;** i**++)** **{**

intString **=** ""**;**

a **=** Integer**.**parseInt**(**ipDecimal**[**i**].**getText**());**

**do** **{**

operador **=** a **%** 2**;**

intString **=** operador **+** intString**;**

a **=** a **/** 2**;**

**}** **while** **(**a **>** 0**);**

**while** **(**intString**.**length**()** **<=** 7**)** **{**

intString **=** "0" **+** intString**;**

**}**

StringBuilder sb **=** **new** StringBuilder**(**provisional**);**

sb**.**reverse**();**

ipBinari**[**i**].**setText**(**intString**);**

**}**

**}**

**public** **void** calcularMascara**()** **{**

**int** a **=** 0**,** operador**;**

String intString **=** ""**;**

**int** mascara **=** Integer**.**parseInt**(**ipMascara**.**getText**());**

**for** **(int** i **=** 0**;** i **<** mascara**;** i**++)** **{**

intString **=** ""**;**

**do** **{**

operador **=** a **%** 2**;**

intString **=** operador **+** intString**;**

a **=** a **/** 2**;**

**}** **while** **(**a **>** 0**);**

intString **=** "1"**;**

**for** **(**i **=** 1**;** i **<** mascara**;** i**++)** **{**

intString **=** intString **+** "1"**;**

**}**

**do** **{**

intString **=** intString **+** "0"**;**

**}** **while** **(**intString**.**length**()** **<** 33**);**

**}**

**for** **(int** j **=** 1**;** j **<=** 4**;** j**++)** **{**

ipMasBinari**[**j**].**setText**(**intString**.**substring**((**j **-** 1**)** **\*** 8**,** **(**j **-** 1**)** **\*** 8 **+** 8**));**

**}**

**}**

**public** **void** calcularXarxa**()** **{**

**for** **(int** i **=** 1**;** i **<=** 4**;** i**++)** **{**

StringBuilder sb **=** **new** StringBuilder**();**

**for** **(int** j **=** 0**;** j **<** 8**;** j**++)** **{**

ipMasBinari**[**i**].**getText**().**substring**(**j**,** j **+** 1**);**

**if** **(**ipMasBinari**[**i**].**getText**().**substring**(**j**,** j **+** 1**).**equals**(**"1"**))** **{**

sb**.**append**(**ipBinari**[**i**].**getText**().**substring**(**j**,** j **+** 1**));**

**}** **else** **{**

sb**.**append**(**"0"**);**

**}**

**}**

xarxaBinari**[**i**].**setText**(**sb**.**toString**());**

**}**

**}**

**private** **static** **boolean** isNumero**(**String cadena**)** **{**

**try** **{**

Integer**.**parseInt**(**cadena**);**

**return** **true;**

**}** **catch** **(**NumberFormatException e**)** **{**

**return** **false;**

**}**

**}**

**private** **void** setTextIpDecimal**()** **{**

**for** **(int** i **=** 1**;** i **<** 5**;** i**++)** **{**

ipDecimal**[**i**]** **=** **new** JTextField**(**3**);**

**}**

**}**

**private** **void** setTextBi**()** **{**

**for** **(int** i **=** 1**;** i **<** 5**;** i**++)** **{**

ipBinari**[**i**]** **=** **new** JTextField**(**8**);**

ipBinari**[**i**].**setEditable**(false);**

**}**

**}**

**private** **void** setTextMascara**()** **{**

**for** **(int** i **=** 1**;** i **<** 5**;** i**++)** **{**

ipMasBinari**[**i**]** **=** **new** JTextField**(**8**);**

ipMasBinari**[**i**].**setEditable**(false);**

**}**

**}**

**private** **void** setTextXarxaBinari**()** **{**

**for** **(int** i **=** 1**;** i **<** 5**;** i**++)** **{**

xarxaBinari**[**i**]** **=** **new** JTextField**(**8**);**

xarxaBinari**[**i**].**setEditable**(false);**

**}**

**}**

**}**

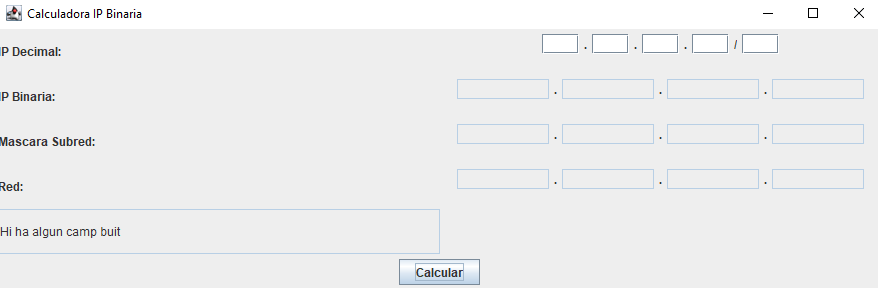
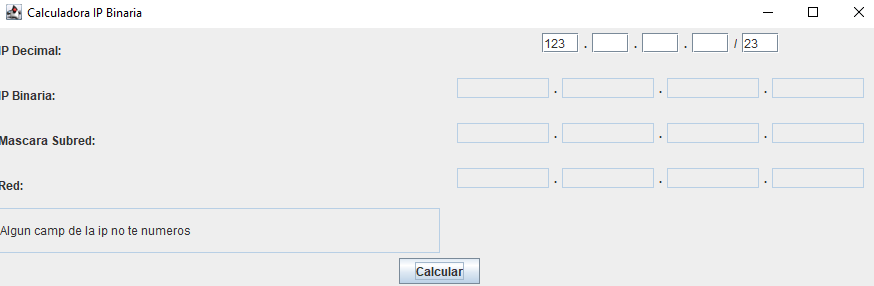
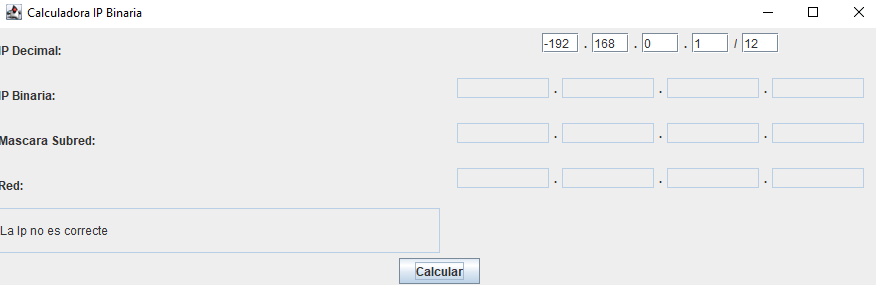
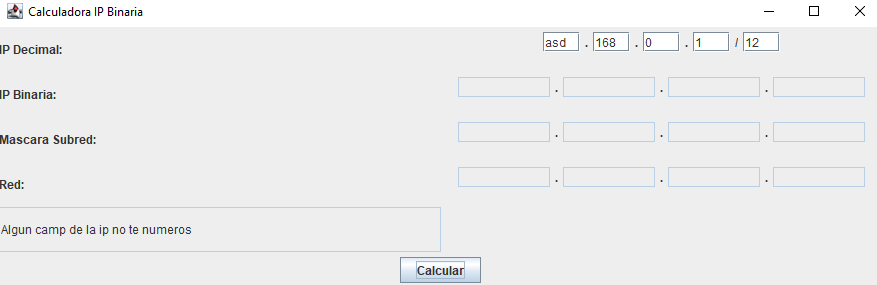
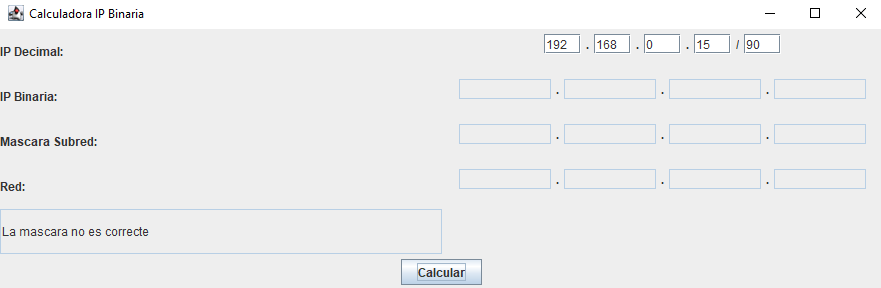
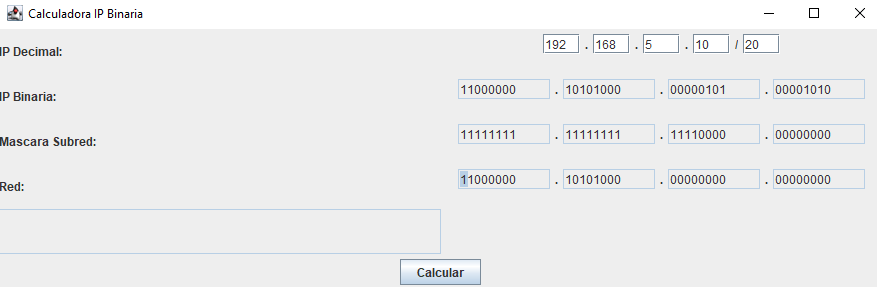
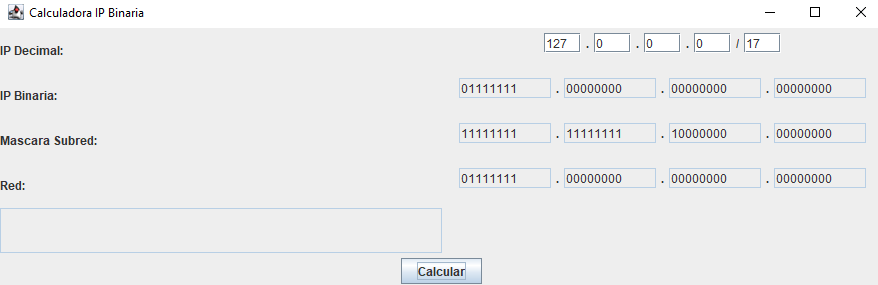
# Documentació del codi:

## En aquest apartat comentaré el codi funció per funció

1. El primer que podem veure al codi es la declaració dels JTextField que farem servir en el codi. Aquests son arrays ja que tractarem les dades com arrays y no com a variables individuals.
2. Després podem veure com declarem la nova finestra en la que treballarem i li donem les seves mides, sota d’això podem veure com criem a les funcions que estan al final del codi que se n’encarreguen d’omplir cada JTextField amb un JTextField de 8 posicions i en el cas dels camps que corresponen al resultat de la calculadora els posem en que no es puguin editar.
3. Sota d’aquest codi (sobre la línia 34) podem veure com declarem un JPanel i li apliquem l’estructura d’un FlowLayout i dins d’aquest panell li anem afegint els camps i els “ . ” o “ / ” depenent del cas.
4. Sobre la línia 80, podem veure com creem el panell que serà la base de l’aplicació i li assignem l’estructura del GridLayout i li afegim uns JLabel i els panells anteriors. Al final d’aquest bloc podem veure com declarem on anirà el camp on es mostraran els erros.
5. Tot seguit podem veure com declarem el botó de calcular i li assignem les mides, i com li diem que quan es cliqui cridi a la funció “EventBoto”
   1. Aquesta funció el que fa es: recorre tota la llargada de la variable “ipDecimal” i aquí dins comprovem:
      1. Si hi ha algun camp buit
      2. Si el que li hem introduït son numeros
      3. Si els numeros introduïts estan entre 255 i 0
      4. Si la mascara esta entre 32 i 0
   2. Totes aquestes comprovacions anteriors es per a fer que l’usuari introdueixi numeros correctes, si ho fa llavors passem a la línia 145 on executem les funcions per a fer la conversió de la IP i la mascara a binari. Si aquesta funció no es compleix assignarem tots els JTextField a “res” per a mostrar en blanc els camps en cas de que hi hagués algún error.

# Proves

## Aquí farem les proves de que l’aplicació funciona correctament.

1. Comprovació de que si no omplim cap camp controlem l’error
2. Comprovació de que si omplim algun camp de la IP i la mascara, ens mostra error
3. Comprovació de que si introduïm numeros negatiu ens dona error
4. Comprovació de que si introduïm lletres ens mostra error
5. Comprovació de que si no posem una mascara correcte ens dona error
6. Comprovacions del funcionament correcte